

## **IN THE SPECIFICATION**

Please replace the paragraph beginning at page 5, line 31, with the following amended version.

--Recent developments in SAN architecture attempt to minimize the need for kernel intervention during a remote memory access operation. FIG. 1 illustrates one such SAN architecture, the InfiniBand<sup>sm</sup> Architecture developed by the Infiniband<sup>sm</sup> Trade Association, the specification for which is hereby incorporated herein by reference (Infiniband is a trademark of the Infiniband Trade Association). The Infiniband architecture defines a first order interconnect technology for interconnecting processor nodes 122, 132, and 142 and IO nodes 112 and 124, as well as hardware transport protocols sufficient to support reliable messaging (send/receive) and memory manipulation semantics without software intervention in the data movement path.--

Please replace the paragraph beginning at page 6, line 17, with the following amended version.

--Host computers 118 at a processor node 122 interface with the fabric 100 through one or more host channel adaptors 106, 126, 136, 146, and 156. The channel adaptors 106, 126, 136, 146, and 156 are hardware components in the processor nodes 122, 132 and IO units 112, 124 that generate and consume data packets. Certain channel adaptors of the Infiniband<sup>sm</sup> architecture are programmable direct memory access engines with special protection features that allow direct memory access operations to be initiated locally or remotely. Further, each of these channel adaptors maintains a translation and protection table ("TPT") that supports memory region translations. The channel adaptors

use the translation and protection table to translate virtual addresses to physical addresses and to validate access rights.--

Please replace the paragraph beginning at page 8, line 4, with the following amended version.

--FIG. 2 is a block diagram illustrating remote memory access translation system. In this embodiment, the system 200 includes a memory region table 201 (or, in the case of the Infiniband<sup>sm</sup> architecture, a translation and protection table). The TPT 201 supports memory region translations and resides on a module external to and in communication with the operating system of a host computer, such as on a channel adapter. The system 200 also includes a memory window table 202 which supports memory window translations. The memory window table also resides outside the operating system of the host computer on a separate module from the memory region table. When a remote direct memory access request is submitted to the system, the request packet contains a remote access key 203 and a virtual address 204. In accordance with an embodiment of the invention, the remote access key is either a region R-key 216 or a window R\_key 208. The packet may also contain a length 205 and operation type (not shown).--

Please replace the paragraph beginning at page 12, line 20, with the following amended version.

--FIG. 7 is a flow chart illustrating a method for binding a memory window to a memory region in accordance with an embodiment of the invention. After a memory region has been registered and a memory window has been allocated,

the user is free to bind the memory window to a memory region. This bind process can be done without invoking privileged code from the operating system, therefore the hardware needs to perform the access security checks. Bind operations may be performed by calling a bind request function, such as that provided by the verbs of the Infiniband<sup>sm</sup> architecture, in process 701. The bind request function creates a work request and then inserts it onto a send queue of a queue pair associated with the bind request. The bind request function passes parameters that include a pointer to a memory window table index and a pointer to a new memory region table index which are read in process 702. The bind request function also passes parameters such as a virtual address, length, a region L\_key, a window R\_key, a memory region handle, a memory window handle, a queue pair handle, and the handle of the hardware device making the bind request.--